(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 977 655 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
26.04.2023 Bulletin 2023/17

(21) Application number: **19729206.3**

(22) Date of filing: **30.05.2019**

(51) International Patent Classification (IPC):
H04L 5/00 *(2006.01)* ***H04L 25/03*** *(2006.01)*
G06N 3/084 *(2023.01)* ***H04L 25/02*** *(2006.01)*
G06N 3/048 *(2023.01)*

(52) Cooperative Patent Classification (CPC):
H04L 25/0228; G06N 3/084; H04L 5/0048; H04L 25/0212; H04L 25/03165; G06N 3/048

(86) International application number:
PCT/EP2019/064146

(87) International publication number:
WO 2020/239232 (03.12.2020 Gazette 2020/49)

(54) LEARNING IN COMMUNICATION SYSTEMS

LERNEN IN KOMMUNIKATIONSSYSTEMEN

APPRENTISSAGE DANS DES SYSTÈMES DE COMMUNICATION

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(43) Date of publication of application:
06.04.2022 Bulletin 2022/14

(73) Proprietor: **Nokia Technologies Oy**
02610 Espoo (FI)

(72) Inventors:
- **HOYDIS, Jakob**
  75014 Paris (FR)
- **AIT AOUDIA, Faycal**
  92210 Saint-Cloud (FR)
- **STARK, Maximilian**
  91300 Massy (FR)

(74) Representative: **Nokia EPO representatives**
Nokia Technologies Oy
Karakaari 7
02610 Espoo (FI)

(56) References cited:
WO-A1-2019/080988

- **FAYCAL AIT AOUDIA ET AL: "End-to-End Learning of Communications Systems Without a Channel Model", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 6 April 2018 (2018-04-06), XP081431831,**
- **FAYCAL AIT AOUDIA ET AL: "Model-free Training of End-to-end Communication Systems", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 14 December 2018 (2018-12-14), XP080993169,**
- **SEBASTIAN D\"ORNER ET AL: "Deep Learning-Based Communication Over the Air", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 11 July 2017 (2017-07-11), XP081279898, DOI: 10.1109/JSTSP.2017.2784180**
- **ERIC JANG ET AL: "Categorical Reparameterization with Gumbel-Softmax", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 3 November 2016 (2016-11-03), XP080729237,**
- **TIMOTHY J O'SHEA ET AL: "An Introduction to Machine Learning Communications Systems", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 2 February 2017 (2017-02-02), XP080746465,**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 3 977 655 B1

Description

Field

[0001] The present specification relates to learning in communication systems.

Background

[0002] A simple communications system includes a transmitter, a transmission channel, and a receiver. The choice of communication scheme used in the transmission of data from the transmitter to the receiver can have a significant impact of the end-to-end performance of such communications systems. Although developments have been made, there remains scope for further developments in this area.

[0003] FAYCAL AIT AOUDIA ET AL, "End-to-End Learning of Communications Systems Without a Channel Model", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, (20180406), proposes a method for end-to-end learning of communications systems through neural network-based autoencoders, which iterates between supervised training of the receiver and reinforcement learning-based training of the transmitter. FAYCAL AIT AOUDIA ET AL, "Model-free Training of End-to-end Communication Systems", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, (20181214), proposes a learning algorithm that enables training of a communication systems with an unknown channel model or with non-differentiable components.

[0004] End-to-end learning in communication systems is also known from WO2019080988.

Summary

[0005] In a first aspect, this specification describes an apparatus according to independent claim 1.

[0006] In a second aspect, this specification describes a method according to independent claim 14.

[0007] In a third aspect, this specification describes a computer program according to independent claim 15

Brief description of the drawings

[0008] Example embodiments will now be described, by way of non-limiting examples, with reference to the following schematic drawings, in which:

FIG. 1 is a block diagram of an example end-to-end communication system in accordance with an example embodiment;
FIG. 2 is a block diagram of an example transmitter that may be used in the communication system of FIG. 1;
FIG. 3 is a block diagram of an example receiver that may be used in the communication system of FIG. 1;
FIG. 4 is a block diagram of a transmission system in accordance with an example embodiment;
FIG. 5 is a block diagram of a transmitter in accordance with an example embodiment;
FIG. 6 is a block diagram of an example receiver module in accordance with an example embodiment;
FIG. 7 is a flow chart showing an algorithm in accordance with an example embodiment;
FIG. 8 is a block diagram of a components of a system in accordance with an exemplary embodiment; and
FIGS. 9a and 9b show tangible media, respectively a removable memory unit and a compact disc (CD) storing computer-readable code which when run by a computer perform operations according to embodiments.

Detailed description

[0009] The scope of protection sought for various embodiments of the invention is set out by the independent claims. The embodiments and features, if any, described in the specification that do not fall under the scope of the independent claims are to be interpreted as examples useful for understanding various embodiments of the invention.

[0010] In the description and drawings, like reference numerals refer to like elements throughout.

[0011] FIG. 1 is a block diagram of an example end-to-end communication system, indicated generally by the reference numeral 1, in accordance with an example embodiment. The system 1 includes a transmitter 2, a channel 4 and a receiver 6. Viewed at a system level, the system 1 converts input bits (b) received at the input to the transmitter 2 into output bits ($\hat{b}$) at the output of the receiver 6. More specifically, the transmitter 2 converts the input bits (b) into transmit symbols (x) for transmission over the channel 4 and the receiver 6 generates the output bits ($\hat{b}$) from symbols (y) received from the channel 4.

[0012] FIG. 2 is a block diagram of an example transmitter 2 that may be used in the communication system 1 described

above. As shown in FIG. 2, the transmitter 2 includes a bit-to-symbols module 12 and a modulator 14. The bits-to-symbols module 12 receives the input data bits (b) and converts those bits into data symbols (s) for transmission. For example, the input data bits (b) may take the form of a data stream, which data stream is packaged for transmission by the bits-to-symbol module 12. The modulator 14 converts the data symbols into transmit symbols (x) in accordance with a modulation scheme. The transmit symbols are then transmitted over the channel 4 and received at the receiver 6 as received symbols (y).

[0013] FIG. 3 is a block diagram of an example receiver 6 that may be used in the communication system 1 described above. As shown in FIG. 3, the receiver 6 includes a demodulator 22 and demapper module 24. The demodulator 22 converts the received symbols (y) into symbol probabilities $p_{\varnothing}(s|\boldsymbol{y})$ and the demapper 24 converts the symbol probabilities into bit probabilities $p(b_i|\boldsymbol{y})$.

[0014] A number of modulation techniques could be used in the implementation of the modulator 14 (and the demodulator 22). These include amplitude shift keying (ASK) in which the amplitude of a carrier signal is modified based on a signal being transmitted and phase shift keying (PSK) in which the phase of a carrier signal is modified based on a signal being transmitted. By way of example, quadrature phase shift keying (QPSK) is a form of phase shift keying in which two bits are modulated at once, selecting one of four possible carrier phases shifts (e.g. 0, +90 degree, 180 degrees, -90 degrees). Such carrier phase and amplitudes are often represented as constellation positions in a complex plane. The skilled person will be aware of many other suitable modulation techniques.

[0015] The choice of modulation scheme used to transmit information such as the communication system 1 has an impact on the end-to-end performance of such communication systems. Moreover, such modulation schemes maybe optimised.

[0016] For communication schemes having multiple constellation points (such as amplitude and/or phase shift keying modulation schemes), the locations of such constellation points in a complex plane may be optimised. Alternatively, or in addition, the relative frequency with which such constellation points are used may be optimised. Those methods are referred to herein as geometric shaping and probabilistic shaping respectively.

[0017] FIG. 4 is a block diagram of a transmission system, indicated generally by the reference numeral 30, in accordance with an example embodiment. The transmission system 30 comprises a transmitter 32, a channel 34 and a receiver 36 (that have some similarities with the transmitter 2, channel 4 and receiver 6 described above).

[0018] As shown in FIG. 4, the transmitter 32 comprises a bits-to-symbol module 37 (similar to the bits-to-symbol module 12 described above), a modulator 38 (similar to the modulator 14) and a shaping device 39. The receiver 36 comprises a demodulator 40 (similar to the demodulator 22) and a demapper module 41 (similar to the demapper module 24).

[0019] As discussed further below, the transmitter 32 maps an input stream of bits onto transmit symbols. The transmitter includes a neural network (or some other function with trainable parameters).

[0020] The bits-to-symbol module 37 receives an incoming bit sequence of size N, denoted by $\boldsymbol{b} = [b_1,...,b_N]$, and maps that bit sequence onto hypersymbols

$$s \in \mathcal{S},$$

such that the frequencies of occurrence of symbols s correspond to a probability distribution $p_{\theta}(s)$ provided by the shaping device 39. Here,

$$\mathcal{S} = \{1, \dots, K\}$$

is the finite and discrete set of hypersymbols. The hypersymbol s is provided to the modulator 38, which maps s onto complex transmit symbols $x \in \mathbb{C}^M$, where M is the number of channel uses.

[0021] The channel 34 of the system 30 takes the complex transmission symbols $\boldsymbol{x} \in \mathbb{C}^M$ and outputs M received samples pooled in $\boldsymbol{y} \in \mathbb{C}^M$. The possibly unknown transition probability $p(\boldsymbol{y}|\boldsymbol{x})$ describes the input-output relation of the channel 34.

[0022] The receiver 36 of the system 30 operates on the sequence $\boldsymbol{y} \in \mathbb{C}^M$ of M samples generated by the channel 34. The demodulator 40 of the receiver 36 maps the received symbols onto symbol probabilities $p_{\phi}(s|\boldsymbol{y})$, and the demapper module 41 outputs bit probabilities $p(b_i|\boldsymbol{y})$.

[0023] FIG. 5 is a block diagram of a transmitter, indicated generally by the reference numeral 50, in accordance with an example embodiment. The transmitter 50 is an example of the transmitter 32 described above; in particular, the transmitter 50 shows an example arrangement for training the transmitter 32 described above.

[0024] The transmitter 50 comprises the bits-to-symbol module 37, the modulator 38 and the shaping device 39 in accordance with an example embodiment. The shaping device 39 may be used during training of the transmitter 50. As described further below, in a normal operation mode, the shaping device 39 may be replaced, for example with a look-up-table providing the appropriate settings.

[0025] The modulator 38 comprises an embedding module 52, a normalization module 54 and real-to-complex conversion module 56. The shaping device 39 comprises a parameters module 60, a summing module 62, a Gumbel sampler 63, a softmax module 64, a softmax τ module 65 and a straight-through estimator 66.

[0026] The bits-to-symbol module 37 maps the incoming bits *b* onto the hypersymbols s. This mapping is implemented such that the frequency with which the symbols are used matches a given probability distribution $p_\theta(s)$. The probability distribution $p_\theta(s)$ is generated by the shaping module 39, as discussed further below.

[0027] The bits-to-symbols module 37 may be implemented in a number of ways. By way of example, a constant composition distribution matching (CCDM) approach maybe used. CCDM involves mapping a vector of N_1 bits to a vector of N_2 symbols, where N_1 and N_2 are typically large numbers. From the set of all possible vectors of symbols of size N_2, one selects the ones such that symbols appear approximately with the desired target probabilities. These selected vectors form the set $\mathcal{G}$. The larger is N_2, the better is the approximation of the target probabilities. N_1 must be such that

$$N_1 \leq \log_2 |\mathcal{G}|.$$

Then, one maps the vectors of bits of size N_1 to the vectors of symbols from $\mathcal{G}$ using arithmetic coding.

[0028] As discussed in detail below, it may be desired to train the system 50 using machine-learning principles, such as backpropagation. The operation performed by the bits-to-symbol module 37 is a discrete and non-differentiable function, and thus, many training techniques used in machine learning are not applicable.

[0029] The shaping device 39 may be implemented as a neural network with parameters θ (or any other trainable function). As suggested in FIG. 5, the shaping device 39 may make use of a straight-through Gumbel-softmax reparameterisation technique. The Gumbel-softmax technique states that samples

$$s \in \mathcal{S}$$

of any discrete probability distribution $p_\theta(s)$ can be approximately sampled by:

$$\tilde{s} = \mathrm{softmax}\left(\frac{g+z}{\tau}\right) \qquad (1)$$

Where

$$\mathrm{softmax}(a) = \left[\frac{exp\, a_i}{\sum_j exp\, a_j}\right]_i \qquad (2)$$

and g_i with

$$i \in \mathcal{S}$$

are samples from a Gumbel (0,1) distribution, and the $\{z_i\}$ with

$$i \in \mathcal{S}$$

are the so-called logits (e.g. the logarithm of the unnormalised probabilityes - the softmax function reverses this process)

computed by a trainable neural network with parameters θ. (The parameter τ is discussed further below.)

[0030] The distribution $p_\theta(s)$ can be retrieved from the logits by $p_\theta = softmax(\boldsymbol{z})$. In the system 50, an example architecture of the neural network generating the logits is depicted. The logits are directly trained in the example, and therefore $\theta = \boldsymbol{z}$. Note that other architectures are possible. For example, the logits could be computer from channel information (such as signal-to-noise ratio) by a neural network.

[0031] By optimizing θ, one optimizes the distribution $p_\theta(s)$.

[0032] If the argmax operator were used instead of the softmax operator in the equation (1) above, this would enable sampling from the exact distribution $p_\theta(s)$. The argmax output would be converted to a one-hot representation, i.e., a vector with a single element set to one and the others set to zero. The index of the element set to one would correspond to the output of the argmax. However, the argmax operator is not differentiable, preventing the optimization of $p_\theta(s)$ by usual stochastic gradient descent (SGD) or similar backpropagation techniques. The softmax function may be used as a differentiable approximation of the argmax operator. The positive parameter τ controls how largely the distribution of the generated samples $\tilde{s}$ deviates from the exact distribution $p_\theta(s)$. Setting τ to low values enables better approximation, but makes the training harder as it increases the sharpness of the softmax τ layer, making backpropagation of the gradient more difficult. A drawback of the softmax function is that it outputs only an approximation of one-hot vectors. To ensure that at training the input of the modulator is always a true one-hot vector, the output of the softmax may be fed into a straight-through estimator 66. A straight-through estimator is a very simple unit which discretizes its input to the closest truly one-hot vector (by setting its largest element equal to one and all other elements to zero), but can be skipped when performing backpropagation (i.e., the discretization is ignored).

[0033] Stochastic gradient descent seeks to optimize the trainable parameters based on backpropagating gradients. Since, in the arrangement described above, only the proposed sampling device but not the bits-to-symbols module 37 is trainable, different submodules are active at training and at deployment. This is depicted by the switch 58 in the system 50.

[0034] During training, the trainable shaping device 39 is active, and the hypersymbols s are generated by this device. In order to learn a distribution $p_\theta(s)$ that maximizes the information rate, the flow of backpropagating gradients is fed into the trainable device and the parameters θ are updated accordingly. Once training is complete, the learned distribution $p_\theta(s)$ is fed into the bits-to-symbols module 37 and the switch 58 is set to "evaluation". Thus, at deployment, the symbols are now generated by the bits-to-symbols module 37, which maps the incoming bit stream onto hypersymbols with the learned frequency $p_\theta(s)$. As noted above, such an arrangement may be implemented by a look-up-table or some similar module.

[0035] The modulator 38 maps the symbols s onto constellation points x in the complex plane. The actual location of these points can either be fixed or jointly optimized with $p_\theta(s)$. If the constellation is also learned, the modulator 38 can implemented as a neural network with trainable parameters (or any other learnable function), and denoted by g_ψ. An example architecture of a trainable modulator is shown in FIG. 5, which includes an embedding layer 52 of dimension $K \times 2M$ and a normalization layer 54 that takes as a second input the learned distribution $p_\theta(s)$ to ensure some power constraint, e.g., $E\left\{||x||_2^2\right\} = M$. Finally, a real-to-complex conversion module 56 maps the $2M$ real outputs to M complex channel symbols. The skilled person will be aware of alternative implementations of the modulator 38 (particularly if the locations of the constellation points are fixed).

[0036] The systems 30 and 50 may be trained using end-to-end learning. With this approach, the transmitter 32, the channel 34 and the receiver 36 may be implemented as a single neural network, and the end-t0-end system trained to reconstruct the input(s) and the output. In some embodiments, a differential channel model may be assumed to be available. Alternative embodiments are possible, for example, in the event that no suitable channel model is available.

[0037] With end-to-end training, a trainable receiver $r_\emptyset$ with parameters Ø implemented as a neural network (or some other trainable function) may be provided.

[0038] FIG. 6 is a block diagram of an example receiver module, indicated generally by the reference numeral 70, in accordance with an example embodiment. The receiver module 70 comprises a complex-to-real conversion module 71, a first dense layer 72, a second dense layer 73 and a softmax layer 74. The provision of two dense layers in the system 70 is an example only; any number of dense layers could be provided.

[0039] The complex-to-real conversion module 71 converts a received vector $\boldsymbol{y} \in \mathbb{C}^M$ into real values. This can be done, for example, by concatenating the real and imaginary parts of a sample to obtain a vector that takes values in $\mathbb{R}^{2M}$.

[0040] The so-obtained vector is fed provided to the dense layers 72 and 73, which dense layers may have different activation functions (e.g. ReLU, tanh, sigmoid, linear etc.). The softmax layer last layer has

$|\mathcal{S}|$,

output dimensions and leverages softmax activation functions to generate a discrete probability mass function $p_\phi(s|\boldsymbol{y})$, whose elements can be interpreted as the probability that the symbol s was transmitted.

[0041] To determine one or more of the trainable function θ (related to probabilistic shaping), φ (related to constellation positions or geometric shaping) and Ø (related to the receiver), stochastic gradient descent (or some similar algorithm) is performed. One possible loss function which could be minimised is the cross-entropy (CE) loss function to which is subtracted the symbols entropy:

$$\mathcal{L} = -E_{s\sim p_\theta(s), y\sim p\left(y\middle|g_\psi(s)\right)}\{\log p_\phi(s|y)\} + \sum_s p_\theta(s)\log p_\theta(s) \qquad (3)$$

[0042] In the equation (3), the expectation is taken over all realisations of the symbols s and channel outputs $\boldsymbol{y}$. Assuming independent and identically distributed (i.i.d.) realisations, the loss function can be estimated by:

$$\mathcal{L}(\theta,\varphi,\phi) \approx -\frac{1}{B}\sum_{i=1}^{B}\log p_\phi\left(s^{(i)}\middle|y^{(i)}\right) + \sum_s p_\theta(s)\log p_\theta(s) \qquad (4)$$

Where B denotes the batch size (i.e. the number of training examples used to approximate the loss). Note that optimisation would not be conducted over φ if the modulator is assumed to have fixed constellations.

[0043] FIG. 7 is a flow chart showing an algorithm, indicated generally by the reference numeral 80, in accordance with an example embodiment.

[0044] The algorithm 80 starts at operation 81, wherein trainable parameters of a transmission system (such as the transmission system discussed above with references to FIGS. 4 and 5) are initialised. The initialisation may take many forms, from example, the initialisation may set the trainable parameters randomly (or pseudo-randomly) or in some arbitrary way; alternatively, the trainable parameters may be set to a predefined starting point or defined according to some algorithm (e.g. on the basis of channel information of the transmission system, such as signal-to-noise ratio).

[0045] The initialisation of parameters may be implemented by the parameters module 60. The parameters module 60 may, for example, set an initial probability function defining a probability that individual data symbols (e.g. constellation points of the transmitter) are output by the transmitter algorithm. The probability function is a discrete function and is not directly differentiable.

[0046] The algorithm moves to operation 82, where training symbols are generated on the basis of a distribution function, wherein the distribution function is differentiable. The distribution function is an approximation (e.g. a differentiable approximation) of the probability function. As noted above, the probability function is typically a discrete and non-differential probability function, which is not conducive to backpropagation techniques that are often employed in machine-learning processes (as discussed further below). Thus, a non-differentiable probability function may be replaced by a differential approximation during training.

[0047] The training symbols may be generated by a Gumbel-softmax algorithm (such as using Gumbel-softmaxreparameterization discussed above). The training symbols, as generated by the shaping device 39 in the training mode, are provided to the modulator 38 and modulated training symbols are transmitted to the receiver 36 over the channel 34.

[0048] As discussed above, the training symbols (in the training mode) may correspond to fixed constellation positions of a modulation scheme implemented by the modulator 38. Similarly, the transmit symbols (in the normal operation mode) may correspond to fixed constellation positions.

[0049] Alternatively, the training symbols, in the training mode (and the transmit symbols in the normal operation mode) may correspond to variable constellation positions of a modulation scheme implemented by said modulator 38. Moreover, the trainable parameters of the transmission system may comprise constellation point positions of said modulation scheme.

[0050] The symbols that are generated and transmitted over said channel in the operation 82 are received by the receiver 36.

[0051] At operation 83, a loss function is generated based on the generated training symbols and the modulated training symbols as received at the receiver of the transmission system. The loss function may take many different forms. For example, the loss function may be related to one or more of block error rate, bit error rate, mutual information and categorical cross-entropy.

[0052] At operation 84, the parameters of the transmission system are updated in order to minimise the loss function, wherein generating updated parameters of the transmission system comprise updating the parameters of the transmitter algorithm (e.g. using machine-learning principles, for example by updating a neural network). Generating updated pa-

rameters of the transmission system may include training the probability function discussed above. Thus, for example, the probability function may be trained in order to set the probabilistic shaping such that the bit error rate (or some other error function) is minimised. Other parameters may also be updated, such as the modulator constellation positions.

[0053] In addition to updating the trainable transmitter algorithm, generating updated parameters of the transmission system in the operation 84 may comprise updating parameters of a trainable receiver algorithm (thereby implementing end-to-end training of the transmission system).

[0054] At operation 85, it is determined whether the algorithm 80 is complete. If so, the algorithm terminates at operation 86; otherwise, the algorithm returns to operation 82 such that the operations 82 to 85 are repeated (i.e. the generating and transmitting of training symbols, the generating a loss function and updating of parameters of the transmission system are repeated).

[0055] The algorithm maybe deemed to be complete (at operation 85) when a condition is reached. Such a condition may take many forms. For example, the operations 82 to 85 may be repeated until a defined performance level is reached. Alternatively, or in addition, the operations 82 to 85 may be repeated until a defined number of iterations has been completed.

[0056] It should also be noted that the batch size B of symbol approximations used in the training made and/or the learning rate (and possibly other parameters of the chosen SGD variant, e,g, ADAM, RMSProp, Momentum) could be optimization parameters of the algorithm 80.

[0057] Once trained, the trainable parameters of the transmission system may be deployed. This may involve updating the trainable parameters of the transmission system using the generated updated parameters of the transmission system. Moreover, the trainable parameters of the transmitter system may comprise one of more of: trainable parameters of the transmitter algorithm, trainable parameters of the modulator and trainable parameters of the receiver.

[0058] A number of modifications to the example embodiments described above are possible. For example, end-to-end training of the transmission systems 30 and 50 could be performed without knowledge of the relevant channel mode, and without requiring a differentiable channel mode, by leveraging principles of reinforcement learning. Moreover, a differential channel model could be learned using the principles of generative adversarial networks. Once such a model has been learned, end-to-end learning could be performed using the learned channel model as a differentiable channel model.

[0059] For completeness, FIG. 8 is a schematic diagram of components of one or more of the modules described previously (e.g. the transmitter or receiver neural networks), which hereafter are referred to generically as processing systems 110. A processing system 110 may have a processor 112, a memory 114 closely coupled to the processor and comprised of a RAM 124 and ROM 122, and, optionally, hardware keys 120 and a display 128. The processing system 110 may comprise one or more network interfaces 118 for connection to a network, e.g. a modem which may be wired or wireless.

[0060] The processor 112 is connected to each of the other components in order to control operation thereof.

[0061] The memory 114 may comprise a non-volatile memory, a hard disk drive (HDD) or a solid state drive (SSD). The ROM 122 of the memory 114 stores, amongst other things, an operating system 125 and may store software applications 126. The RAM 124 of the memory 114 is used by the processor 112 for the temporary storage of data. The operating system 125 may contain code which, when executed by the processor, implements aspects of the algorithm 80.

[0062] The processor 112 may take any suitable form. For instance, it may be a microcontroller, plural microcontrollers, a processor, or plural processors.

[0063] The processing system 110 may be a standalone computer, a server, a console, or a network thereof.

[0064] In some embodiments, the processing system 110 may also be associated with external software applications. These may be applications stored on a remote server device and may run partly or exclusively on the remote server device. These applications may be termed cloud-hosted applications. The processing system 110 may be in communication with the remote server device in order to utilize the software application stored there.

[0065] FIGS. 9A and 9B show tangible media, respectively a removable memory unit 165 and a compact disc (CD) 168, storing computer-readable code which when run by a computer may perform methods according to embodiments described above. The removable memory unit 165 may be a memory stick, e.g. a USB memory stick, having internal memory 166 storing the computer-readable code. The memory 166 may be accessed by a computer system via a connector 167. The CD 168 maybe a CD-ROM or a DVD or similar. Other forms of tangible storage media may be used.

[0066] Embodiments of the present invention may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. The software, application logic and/or hardware may reside on memory, or any computer media. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a "memory" or "computer-readable medium" may be any non-transitory media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

[0067] Reference to, where relevant, "computer-readable storage medium", "computer program product", "tangibly

embodied computer program" etc., or a "processor" or "processing circuitry" etc. should be understood to encompass not only computers having differing architectures such as single/ multi-processor architectures and sequencers/parallel architectures, but also specialised circuits such as field programmable gate arrays FPGA, application specify circuits ASIC, signal processing devices and other devices. References to computer program, instructions, code etc. should be understood to express software for a programmable processor firmware such as the programmable content of a hardware device as instructions for a processor or configured or configuration settings for a fixed function device, gate array, programmable logic device, etc.

[0068] As used in this application, the term "circuitry" refers to all of the following: (a) hardware-only circuit implementations (such as implementations in only analogue and/or digital circuitry) and (b) to combinations of circuits and software (and/ or firmware), such as (as applicable): (i) to a combination of processor(s) or (ii) to portions of processor(s)/ software (including digital signal processor(s)), software, and memory(ies) that work together to cause an apparatus, such as a server, to perform various functions) and (c) to circuits, such as a microprocessor(s) or a portion of a microprocessor(s), that require software or firmware for operation, even if the software or firmware is not physically present.

[0069] If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described functions may be optional or may be combined. Similarly, it will also be appreciated that the flow diagram of FIG. 7 is an example only and that various operations depicted therein maybe omitted, reordered and/or combined.

[0070] It will be appreciated that the above described example embodiments are purely illustrative and are not limiting on the scope of the invention. Other variations and modifications will be apparent to persons skilled in the art upon reading the present specification.

[0071] Moreover, the disclosure of the present application should be understood to include any novel features or any novel combination of features either explicitly or implicitly disclosed herein or any generalization thereof and during the prosecution of the present application or of any application derived therefrom, new claims may be formulated to cover any such features and/or combination of such features.

[0072] Although various aspects of the invention are set out in the independent claims, other aspects of the invention comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims.

[0073] It is also noted herein that while the above describes various examples, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications which may be made without departing from the scope of the present invention as defined in the appended claims.

Claims

1. A transmitter (32) for a transmission system (30), wherein the transmission system (30) comprises the transmitter (32), a channel (34) and a receiver (36), wherein the transmitter (32) includes a transmitter algorithm for converting one or more inputs into one or more data symbols and a modulator (38) for converting said data symbols into transmit symbols in accordance with a modulation scheme, wherein the transmitter (32) further comprises:

   means for initialising trainable parameters of the transmission system (30), wherein the trainable parameters of the transmission system include a probability function defining a probability that individual data symbols are output by the transmitter algorithm;
   means for generating training symbols on the basis of a distribution function, wherein the distribution function is differentiable, and
   wherein the distribution function is an approximation of the probability function;
   means for transmitting modulated training symbols to the receiver (36) over the channel (34) in a training mode;
   means for generating a loss function based on the generated training symbols and the modulated training symbols as received at the receiver (36) of the transmission system (30);
   means for generating updated parameters of the transmission system (30) in order to minimise the loss function, wherein generating updated parameters of the transmission system comprise updating the parameters of the transmitter algorithm; and
   means for repeating the generating the training symbols, generating the loss function and updating parameters of the transmission system until a first condition is reached.

2. The transmitter (32) as claimed in claim 1, wherein the training symbols are generated by a Gumbel-softmax algorithm.

3. The transmitter (32) as claimed in claim 1 or claim 2, wherein said transmit symbols correspond to fixed constellation

positions of a modulation scheme implemented by said modulator.

4. The transmitter (32) as claimed in claim 1 or claim 2, wherein said transmit symbols correspond to variable constellation positions of a modulation scheme implemented by said modulator, wherein the trainable parameters of the transmission system comprise constellation point positions of said modulation scheme.

5. The transmitter (32) as claimed in any one of the preceding claims, wherein the transmitter (32) is configured to transmit said transmit symbols over said channel (34) to the receiver (36) in an operational mode and to transmit said modulated training symbols over the channel (36) to the receiver (36) in a training mode.

6. The transmitter (32) as claimed in any one of the preceding claims, wherein generating updated parameters of the transmission system comprises updating parameters of a trainable receiver algorithm.

7. The transmitter (32) as claimed in any one of the preceding claims, further comprising means for updating the trainable parameters of the transmission system (30) using the generated updated parameters of the transmission system, wherein the trainable parameters of the transmission system comprise one of more of: trainable parameters of the transmitter algorithm; trainable parameters of the modulator; and trainable parameters of the receiver.

8. The transmitter (32) as claimed in any one of the preceding claims, wherein the probability function and the distribution function are at least partially dependent on channel information of the transmission system.

9. The transmitter (32) as claimed in any one of the preceding claims, wherein the first condition comprises a defined performance level, or the first condition comprises a defined number of iterations.

10. The transmitter (32) as claimed in any one of the preceding claims, wherein said transmitter algorithm is implemented as a look-up table in an operational mode.

11. The transmitter (32) as claimed in any one of the preceding claims, wherein the loss function is related to one or more of block error rate, bit error rate, mutual information and categorical cross-entropy.

12. The transmitter (32) as claimed in any one of the preceding claims, wherein the means for generating updated parameters of the transmission system (30) optimises one or more of a batch size of symbol approximations used in the training mode and a learning rate.

13. The transmitter (32) as claimed in any one of the preceding claims, wherein the transmitter (32) comprises a transmitter neural network configured to implement said transmitter algorithm.

14. A method comprising:

initialising (81) trainable parameters of a transmission system (30), wherein the transmission system (30) comprises a transmitter (32), a channel (34) and a receiver (36), wherein the transmitter (32) includes a transmitter algorithm for converting one or more inputs into one or more data symbols and a modulator for converting said data symbols into transmit symbols in accordance with a modulation scheme, wherein the trainable parameters of the transmission system include a probability function defining a probability that individual data symbols are output by the transmitter algorithm;
generating (82) training symbols on the basis of a distribution function, wherein the distribution function is differentiable, and wherein the distribution function is an approximation of the probability function;
transmitting (82) modulated training symbols to the receiver over the channel in a training mode;
generating (83) a loss function based on the generated training symbols and the modulated training symbols as received at the receiver of the transmission system;
generating (84) updated parameters of the transmission system in order to minimise the loss function, wherein generating updated parameters of the transmission system comprise updating the parameters of the transmitter algorithm; and
repeating (85) the generating the training symbols, generating the loss function and updating parameters of the transmission system until a first condition is reached.

15. A computer program comprising instructions which, when the program is executed by a computer, cause the computer to perform at least the following:

Initialise (81) trainable parameters of a transmission system (30), wherein the transmission system (30) comprises a transmitter (32), a channel (34) and a receiver (36), wherein the transmitter (32) includes a transmitter algorithm for converting one or more inputs into one or more data symbols and a modulator for converting said data symbols into transmit symbols in accordance with a modulation scheme, wherein the trainable parameters of the transmission system include a probability function defining a probability that individual data symbols are output by the transmitter algorithm;
generate (82) training symbols on the basis of a distribution function, wherein the distribution function is differentiable, and wherein the distribution function is an approximation of the probability function;
transmit (82) modulated symbol to the receiver over the channel in a training mode;
generate (83) a loss function based on the generated symbol and the modulated symbol as received at the receiver of the transmission system;
generate (84) updated parameters of the transmission system in order to minimise the loss function, wherein generating updated parameters of the transmission system comprise updating the parameters of the transmitter algorithm; and
repeat (85) the generating the symbol approximations, generating the loss function and updating parameters of the transmission system until a first condition is reached.

Patentansprüche

1. Sender (32) für ein Übertragungssystem (30), wobei das Übertragungssystem (30) den Sender (32), einen Kanal (34) und einen Empfänger (36) umfasst, wobei der Sender (32) einen Senderalgorithmus zum Umwandeln von einer oder mehreren Eingaben in ein oder mehrere Datensymbole und einen Modulator (38) zum Umwandeln der Datensymbole gemäß einem Modulationsschema in Sendesymbole beinhaltet, wobei der Sender (32) ferner Folgendes umfasst:

   Mittel zum Initialisieren von trainierbaren Parametern des Übertragungssystems (30), wobei die trainierbaren Parameter des Übertragungssystems eine Wahrscheinlichkeitsfunktion beinhalten, die eine Wahrscheinlichkeit definiert, dass einzelne Datensymbole vom Senderalgorithmus ausgegeben werden;
   Mittel zum Erzeugen von Trainingssymbolen auf Basis einer Verteilungsfunktion, wobei die Verteilungsfunktion differenzierbar ist, und wobei die Verteilungsfunktion eine Annäherung der Wahrscheinlichkeitsfunktion ist;
   Mittel zum Übertragen von modulierten Trainingssymbolen in einem Trainingsmodus über den Kanal (34) zum Empfänger (36);
   Mittel zum Erzeugen einer Verlustfunktion auf Basis der erzeugten Trainingssymbole und der modulierten Trainingssymbole, die am Empfänger (36) des Übertragungssystems (30) empfangen werden;
   Mittel zum Erzeugen von aktualisierten Parametern des Übertragungssystems (30), um die Verlustfunktion zu minimieren, wobei das Erzeugen von aktualisierten Parametern des Übertragungssystems das Aktualisieren der Parameter des Senderalgorithmus umfasst; und
   Mittel zum Wiederholen des Erzeugens der Trainingssymbole, des Erzeugens der Verlustfunktion und des Aktualisierens von Parametern des Übertragungssystems, bis ein erster Zustand erreicht ist.

2. Sender (32) nach Anspruch 1, wobei die Trainingssymbole von einem Gumbel-softmax-Algorithmus erzeugt werden.

3. Sender (32) nach Anspruch 1 oder Anspruch 2, wobei die Sendesymbole festen Konstellationspositionen eines vom Modulator implementierten Modulationsschemas entsprechen.

4. Sender (32) nach Anspruch 1 oder Anspruch 2, wobei die Sendesymbole variablen Konstellationspositionen eines vom Modulator implementierten Modulationsschemas entsprechen, wobei die trainierbaren Parameter des Übertragungssystems Konstellationspunktpositionen des Modulationsschemas umfassen.

5. Sender (32) nach einem der vorhergehenden Ansprüche, wobei der Sender (32) dazu ausgelegt ist, die Sendesymbole in einem Betriebsmodus über den Kanal (34) zum Empfänger (36) zu übertragen und die modulierten Trainingssymbole in einem Trainingsmodus über den Kanal (36) zum Empfänger (36) zu übertragen.

6. Sender (32) nach einem der vorhergehenden Ansprüche, wobei das Erzeugen von aktualisierten Parametern des Übertragungssystems das Aktualisieren von Parametern eines trainierbaren Empfängeralgorithmus umfasst.

7. Sender (32) nach einem der vorhergehenden Ansprüche, der ferner Mittel zum Aktualisieren der trainierbaren

Parameter des Übertragungssystems (30) unter Verwendung der erzeugten aktualisierten Parameter des Übertragungssystems umfasst, wobei die trainierbaren Parameter des Übertragungssystems eines oder mehreres von Folgendem umfassen: trainierbare Parameter des Senderalgorithmus; trainierbare Parameter des Modulators und trainierbare Parameter des Empfängers.

8. Sender (32) nach einem der vorhergehenden Ansprüche, wobei die Wahrscheinlichkeitsfunktion und die Verteilungsfunktion mindestens teilweise von Kanalinformationen des Übertragungssystems abhängig sind.

9. Sender (32) nach einem der vorhergehenden Ansprüche, wobei der erste Zustand ein definiertes Leistungsniveau umfasst oder der erste Zustand eine definierte Anzahl von Wiederholungen umfasst.

10. Sender (32) nach einem der vorhergehenden Ansprüche, wobei der Senderalgorithmus als eine Nachschlagetabelle in einem Betriebsmodus implementiert wird.

11. Sender (32) nach einem der vorhergehenden Ansprüche, wobei die Verlustfunktion eines oder mehreres von einer Blockfehlerrate, einer Bitfehlerrate, gegenseitigen Informationen und einer kategorischen Kreuzentropie betrifft.

12. Sender (32) nach einem der vorhergehenden Ansprüche, wobei die Mittel zum Erzeugen von aktualisierten Parametern des Übertragungssystems (30) eines oder mehreres von einer Stapelgröße von Symbolannäherungen, die im Trainingsmodus verwendet werden, und einer Lernrate optimieren.

13. Sender (32) nach einem der vorhergehenden Ansprüche, wobei der Sender (32) ein neuronales Sendernetzwerk umfasst, das dazu ausgelegt ist, den Senderalgorithmus zu implementieren.

14. Verfahren, das Folgendes umfasst:

Initialisieren (81) von trainierbaren Parametern eines Übertragungssystems (30), wobei das Übertragungssystem (30) einen Sender (32), einen Kanal (34) und einen Empfänger (36) umfasst, wobei der Sender (32) einen Senderalgorithmus zum Umwandeln von einer oder mehreren Eingaben in ein oder mehrere Datensymbole und einen Modulator zum Umwandeln der Datensymbole gemäß einem Modulationsschema in Sendesymbole beinhaltet, wobei die trainierbaren Parameter des Übertragungssystems eine Wahrscheinlichkeitsfunktion beinhalten, die eine Wahrscheinlichkeit definiert, dass einzelne Datensymbole vom Senderalgorithmus ausgegeben werden;
Erzeugen (82) von Trainingssymbolen auf Basis einer Verteilungsfunktion, wobei die Verteilungsfunktion differenzierbar ist und wobei die Verteilungsfunktion eine Annäherung der Wahrscheinlichkeitsfunktion ist;
Übertragen (82) von modulierten Trainingssymbolen in einem Trainingsmodus über den Kanal zum Empfänger;
Erzeugen (83) einer Verlustfunktion auf Basis der erzeugten Trainingssymbole und der modulierten Trainingssymbole, die am Empfänger des Übertragungssystems empfangen werden;
Erzeugen (84) von aktualisierten Parametern des Übertragungssystems, um die Verlustfunktion zu minimieren, wobei das Erzeugen von aktualisierten Parametern des Übertragungssystems das Aktualisieren der Parameter des Senderalgorithmus umfasst; und
Wiederholen (85) des Erzeugens der Trainingssymbole, des Erzeugens der Verlustfunktion und des Aktualisierens von Parametern des Übertragungssystems, bis ein erster Zustand erreicht ist.

15. Computerprogramm, das Anweisungen umfasst, die, wenn das Programm von einem Computer ausgeführt wird, den Computer veranlassen, mindestens Folgendes durchzuführen:

Initialisieren (81) von trainierbaren Parametern eines Übertragungssystems (30), wobei das Übertragungssystem (30) einen Sender (32), einen Kanal (34) und einen Empfänger (36) umfasst, wobei der Sender (32) einen Senderalgorithmus zum Umwandeln von einer oder mehreren Eingaben in ein oder mehrere Datensymbole und einen Modulator zum Umwandeln der Datensymbole gemäß einem Modulationsschema in Sendesymbole beinhaltet, wobei die trainierbaren Parameter des Übertragungssystems eine Wahrscheinlichkeitsfunktion beinhalten, die eine Wahrscheinlichkeit definiert, dass einzelne Datensymbole vom Senderalgorithmus ausgegeben werden;
Erzeugen (82) von Trainingssymbolen auf Basis einer Verteilungsfunktion, wobei die Verteilungsfunktion differenzierbar ist, und wobei die Verteilungsfunktion eine Annäherung der Wahrscheinlichkeitsfunktion ist;
Übertragen (82) eines modulierten Symbols in einem Trainingsmodus über den Kanal zum Empfänger;
Erzeugen (83) einer Verlustfunktion auf Basis des erzeugten Symbols und des modulierten Symbols, die am

Empfänger des Übertragungssystems empfangen werden;
Erzeugen (84) von aktualisierten Parametern des Übertragungssystems, um die Verlustfunktion zu minimieren, wobei das Erzeugen von aktualisierten Parametern des Übertragungssystems das Aktualisieren der Parameter des Senderalgorithmus umfasst; und
Wiederholen (85) des Erzeugens der Symbolannäherungen, des Erzeugens der Verlustfunktion und des Aktualisierens von Parametern des Übertragungssystems, bis ein erster Zustand erreicht ist.

Revendications

1. Émetteur (32) pour un système de transmission (30), dans lequel le système de transmission (30) comprend l'émetteur (32), un canal (34) et un récepteur (36), dans lequel l'émetteur (32) comporte un algorithme d'émetteur pour convertir une ou plusieurs entrées en un ou plusieurs symboles de données et un modulateur (38) pour convertir lesdits symboles de données en symboles de transmission conformément à un schéma de modulation, dans lequel l'émetteur (32) comprend en outre :

   un moyen pour initialiser des paramètres entraînables du système de transmission (30), dans lequel les paramètres entraînables du système de transmission comportent une fonction de probabilité définissant une probabilité que des symboles de données individuels soient délivrés par l'algorithme d'émetteur ;
   un moyen pour générer des symboles d'entraînement sur la base d'une fonction de distribution, dans lequel la fonction de distribution est différentiable, et dans lequel la fonction de distribution est une approximation de la fonction de probabilité ;
   un moyen pour transmettre des symboles d'entraînement modulés au récepteur (36) sur le canal (34) dans un mode d'entraînement ;
   un moyen pour générer une fonction de perte sur la base des symboles d'entraînement générés et des symboles d'entraînement modulés à leur réception par le récepteur (36) du système de transmission (30) ;
   un moyen pour générer des paramètres mis à jour du système de transmission (30) afin de minimiser la fonction de perte, dans lequel la génération de paramètres mis à jour du système de transmission comprend la mise à jour des paramètres de l'algorithme d'émetteur ; et
   un moyen pour répéter la génération des symboles d'entraînement, générer la fonction de perte et mettre à jour les paramètres du système de transmission jusqu'à ce qu'une première condition soit atteinte.

2. Émetteur (32) selon la revendication 1, dans lequel les symboles d'entraînement sont générés par un algorithme Gumbel-softmax.

3. Émetteur (32) selon la revendication 1 ou la revendication 2, dans lequel lesdits symboles de transmission correspondent à des positions de constellation fixes d'un schéma de modulation mis en œuvre par ledit modulateur.

4. Émetteur (32) selon la revendication 1 ou la revendication 2, dans lequel lesdits symboles de transmission correspondent à des positions de constellation variables d'un schéma de modulation mis en œuvre par ledit modulateur, dans lequel les paramètres entraînables du système de transmission comprennent des positions de points de constellation dudit schéma de modulation.

5. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel l'émetteur (32) est configuré pour transmettre lesdits symboles de transmission sur ledit canal (34) au récepteur (36) dans un mode de fonctionnement et pour transmettre lesdits symboles d'entraînement modulés sur le canal (36) au récepteur (36) dans un mode d'entraînement.

6. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel la génération de paramètres mis à jour du système de transmission comprend la mise à jour de paramètres d'un algorithme de récepteur entraînable.

7. Émetteur (32) selon l'une quelconque des revendications précédentes, comprenant en outre un moyen pour mettre à jour les paramètres entraînables du système de transmission (30) en utilisant les paramètres mis à jour générés du système de transmission, dans lequel les paramètres entraînables du système de transmission comprennent un ou plusieurs parmi : des paramètres entraînables de l'algorithme d'émetteur ; des paramètres entraînables du modulateur ; et des paramètres entraînables du récepteur.

8. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel la fonction de probabilité et la fonction de distribution dépendent au moins partiellement d'informations de canal du système de transmission.

9. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel la première condition comprend un niveau de performance défini, ou la première condition comprend un nombre défini d'itérations.

10. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel ledit algorithme d'émetteur est mis en œuvre sous forme de table de consultation dans un mode de fonctionnement.

11. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel la fonction de perte est liée à un ou plusieurs parmi un taux d'erreur sur les blocs, un taux d'erreur sur les bits, des informations mutuelles et une entropie croisée catégorielle.

12. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel le moyen de génération de paramètres mis à jour du système de transmission (30) optimise un ou plusieurs parmi une taille de lot d'approximations de symboles utilisés dans le mode d'entraînement et un taux d'apprentissage.

13. Émetteur (32) selon l'une quelconque des revendications précédentes, dans lequel l'émetteur (32) comprend un réseau neuronal d'émetteur configuré pour mettre en œuvre ledit algorithme d'émetteur.

14. Procédé comprenant :

l'initialisation (81) de paramètres entraînables d'un système de transmission (30), dans lequel le système de transmission (30) comprend un émetteur (32), un canal (34) et un récepteur (36), dans lequel l'émetteur (32) comporte un algorithme d'émetteur pour convertir une ou plusieurs entrées en un ou plusieurs symboles de données et un modulateur pour convertir lesdits symboles de données en symboles de transmission conformément à un schéma de modulation, dans lequel les paramètres entraînables du système de transmission comportent une fonction de probabilité définissant une probabilité que des symboles de données individuels soient délivrés par l'algorithme d'émetteur ;
la génération (82) de symboles d'entraînement sur la base d'une fonction de distribution,
dans lequel la fonction de distribution est différentiable, et
dans lequel la fonction de distribution est une approximation de la fonction de probabilité ;
la transmission (82) de symboles d'entraînement modulés au récepteur sur le canal dans un mode d'entraînement ;
la génération (83) d'une fonction de perte sur la base des symboles d'entraînement générés et des symboles d'entraînement modulés à leur réception par le récepteur du système de transmission ;
la génération (84) de paramètres mis à jour du système de transmission afin de minimiser la fonction de perte, dans lequel la génération de paramètres mis à jour du système de transmission comprend la mise à jour des paramètres de l'algorithme d'émetteur ; et
la répétition (85) de la génération des symboles d'entraînement, la génération de la fonction de perte et la mise à jour de paramètres du système de transmission jusqu'à ce qu'une première condition soit atteinte.

15. Programme informatique comprenant des instructions qui, lorsque le programme est exécuté par un ordinateur, amènent l'ordinateur à réaliser au moins ce qui suit :

l'initialisation (81) de paramètres entraînables d'un système de transmission (30), dans lequel le système de transmission (30) comprend un émetteur (32), un canal (34) et un récepteur (36), dans lequel l'émetteur (32) comporte un algorithme d'émetteur pour convertir une ou plusieurs entrées en un ou plusieurs symboles de données et un modulateur pour convertir lesdits symboles de données en symboles de transmission conformément à un schéma de modulation, dans lequel les paramètres entraînables du système de transmission comportent une fonction de probabilité définissant une probabilité que des symboles de données individuels soient délivrés par l'algorithme d'émetteur ;
la génération (82) de symboles d'entraînement sur la base d'une fonction de distribution, dans lequel la fonction de distribution est différentiable, et dans lequel la fonction de distribution est une approximation de la fonction de probabilité ;
la transmission (82) d'un symbole modulé au récepteur sur le canal dans un mode d'entraînement ;
la génération (83) d'une fonction de perte sur la base du symbole généré et du symbole modulé à leur réception par le récepteur du système de transmission ;

la génération (84) de paramètres mis à jour du système de transmission afin de minimiser la fonction de perte, dans lequel la génération de paramètres mis à jour du système de transmission comprend la mise à jour des paramètres de l'algorithme d'émetteur ; et

la répétition (85) de la génération des approximations de symboles, de la génération de la fonction de perte et de la mise à jour des paramètres du système de transmission jusqu'à ce qu'une première condition soit atteinte.

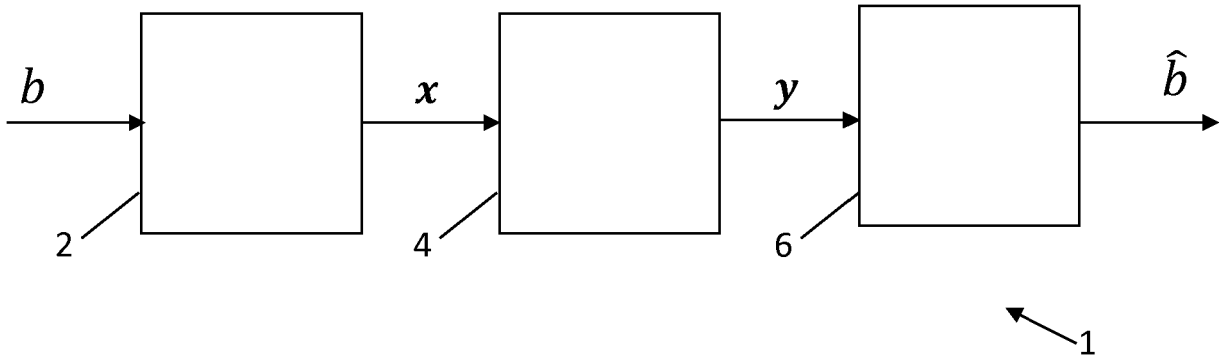

Fig. 1

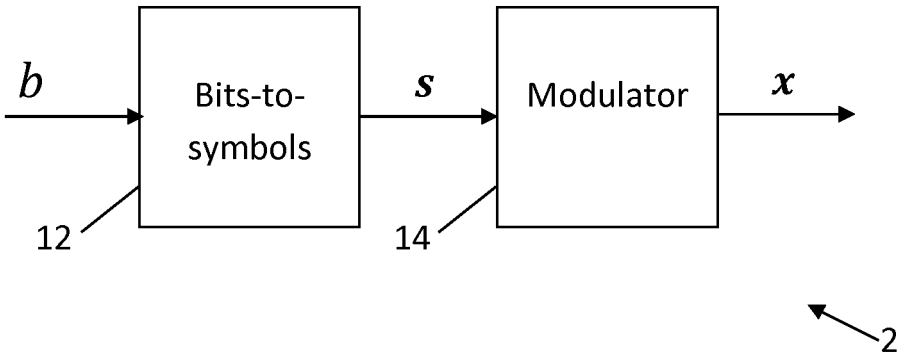

Fig. 2

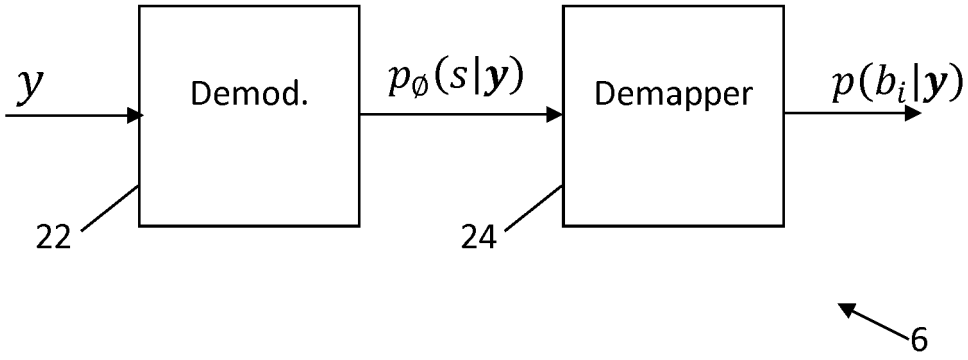

Fig. 3

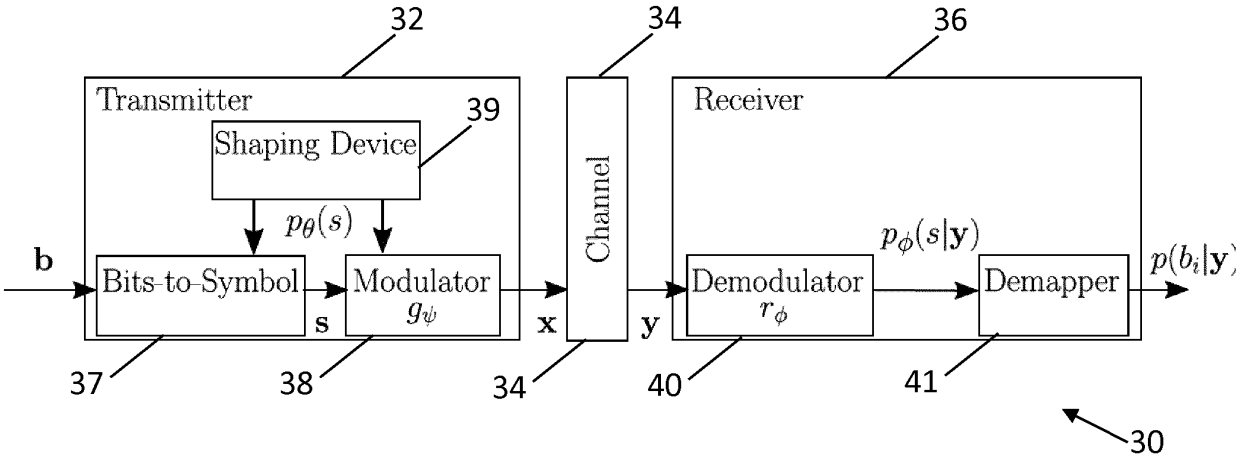
32
34
36
Transmitter
39
Shaping Device
$p_\theta(s)$
b
Bits-to-Symbol
Modulator
g_ψ
s
x
Channel
y
Receiver
$p_\phi(s|\mathbf{y})$
Demodulator
r_ϕ
Demapper
$p(b_i|\mathbf{y})$
37
38
34
40
41
30

Fig. 4

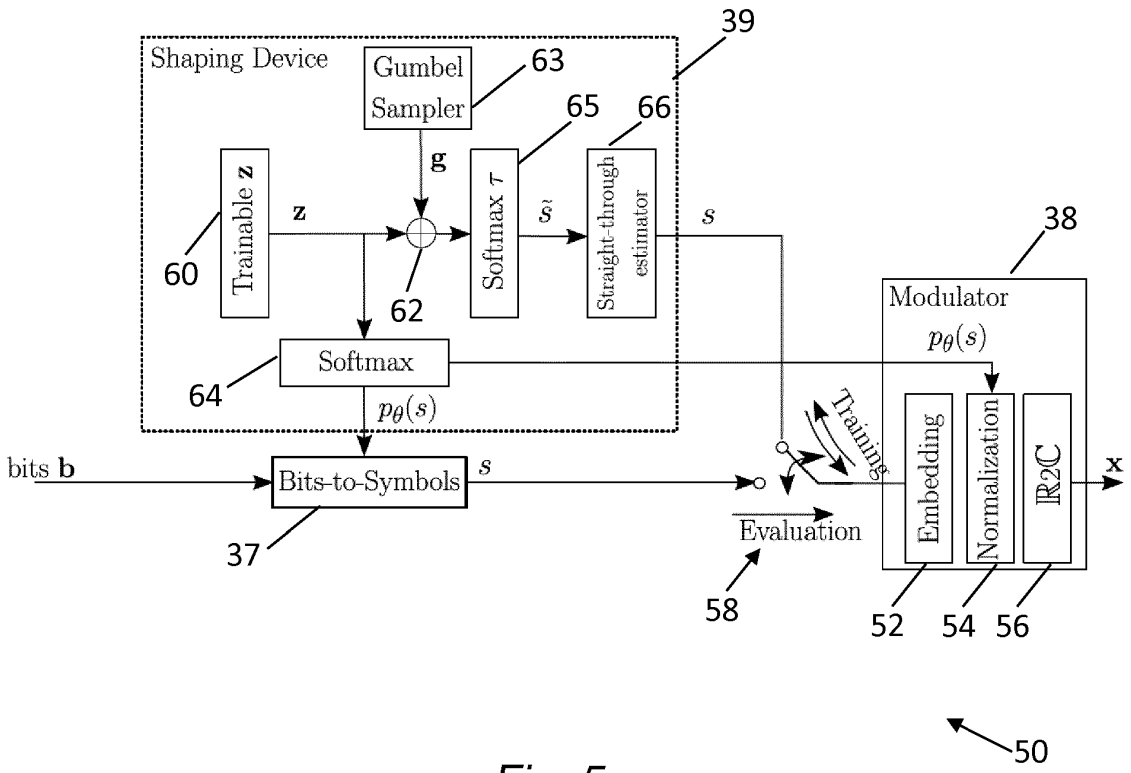
39
Shaping Device
Gumbel Sampler
63
65
66
g
Trainable z
z
60
Softmax τ
s_i
Straight-through estimator
s
62
38
Modulator
$p_\theta(s)$
Softmax
64
$p_\theta(s)$
Training
Embedding
Normalization
$\mathbb{R}^2\mathbb{C}$
x
bits b
Bits-to-Symbols
s
Evaluation
37
58
52
54
56
50

Fig. 5

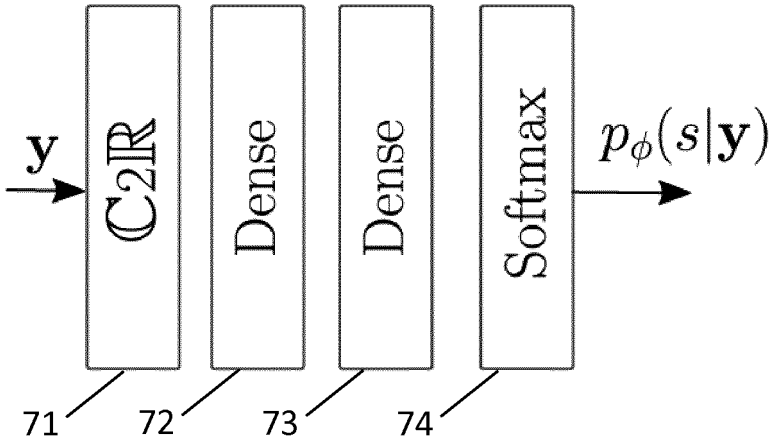

Fig. 6

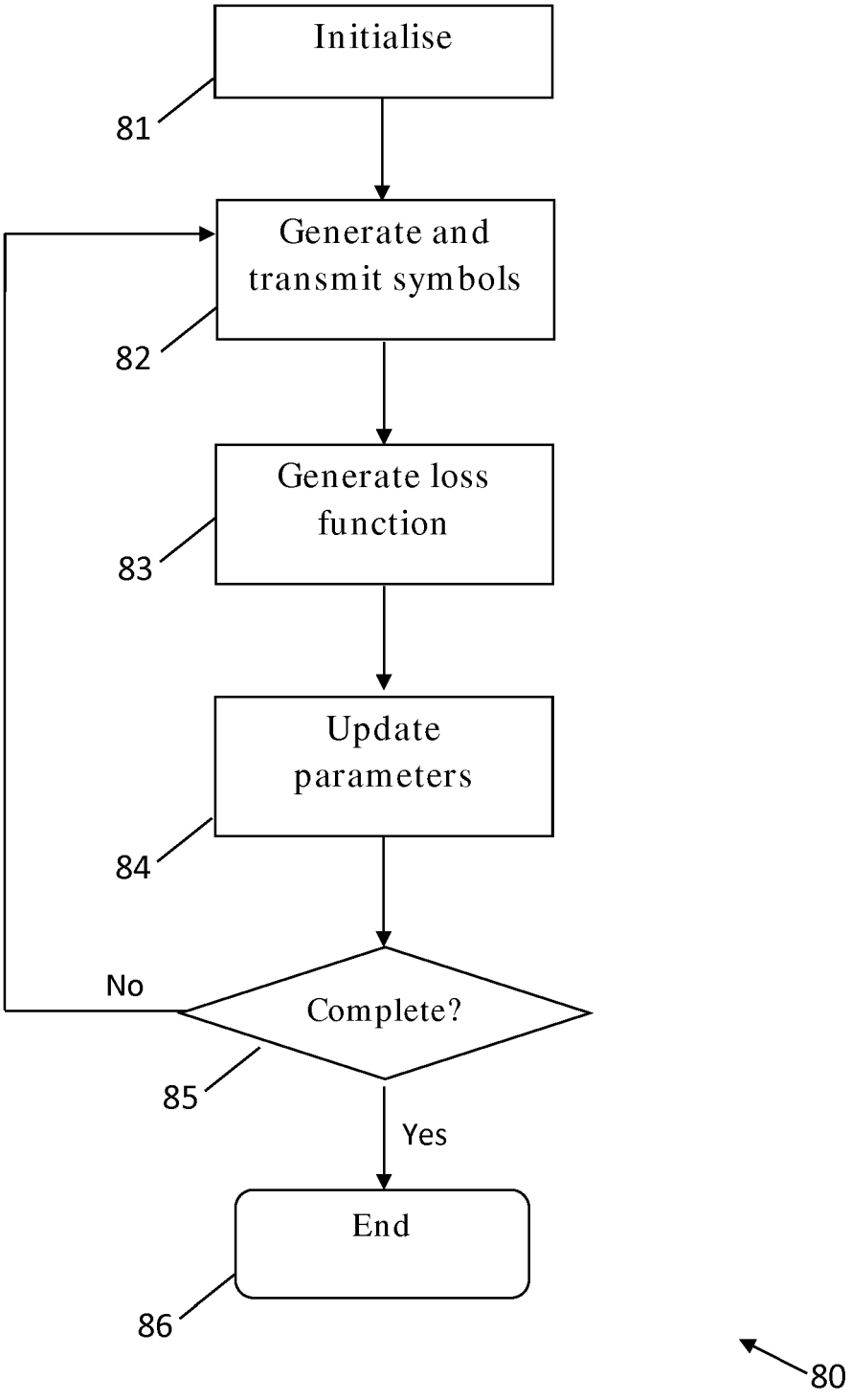
Initialise
81
Generate and transmit symbols
82
Generate loss function
83
Update parameters
84
No
Complete?
85
Yes
End
86
80

Fig. 7

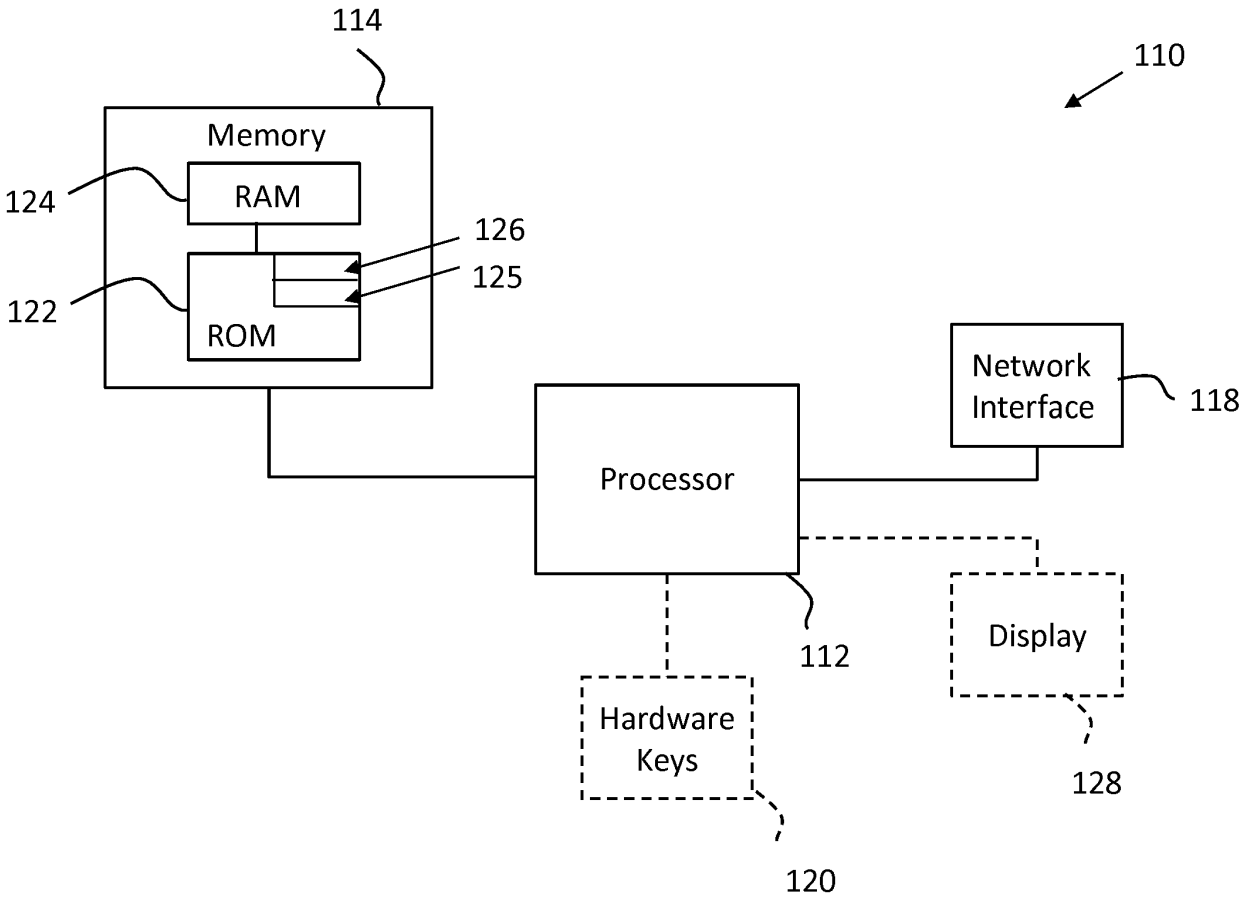
114
110
Memory
RAM
124
126
125
122
ROM
Network
Interface
118
Processor
112
Display
Hardware
Keys
120
128

Fig. 8

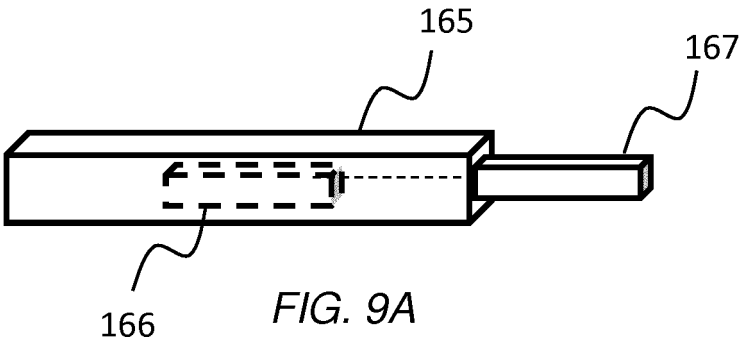

FIG. 9A

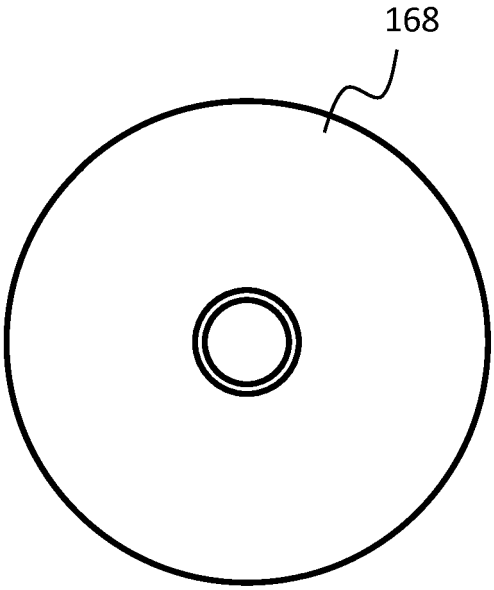

FIG. 9B

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2019080988 A **[0004]**

Non-patent literature cited in the description

- End-to-End Learning of Communications Systems Without a Channel Model. **FAYCAL AIT AOUDIA et al.** ARXIV.ORG. CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, 06 April 2018 **[0003]**
- Model-free Training of End-to-end Communication Systems. **FAYCAL AIT AOUDIA et al.** ARXIV.ORG. CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, 14 December 2018 **[0003]**